

AMENDMENTS TO THE CLAIMS

1. (Currently amended) Method for updating software present in a first version (~~V1~~) in a device (~~40~~), by receiving and executing a delta file (~~D12~~) for upgrading said first version to a second version (~~V2~~), which software is divided and stored in a number of memory blocks (~~201-204~~) defined in an address space (~~211~~) of a physical memory (~~48~~) in the device, comprising the steps of:

- defining an extra memory block (~~212~~) associated with said number of memory blocks, initially being placed before a first memory block located at one end of the memory space;
- erasing said extra memory block;
- writing updated data for the first memory block on said extra memory block;
- moving the extra memory block one block forward; and
- processing all of said number of memory blocks one by one by
 - erasing said extra memory block,
 - writing updated data for the memory block adjacent to and after the extra memory block on the extra memory block,
 - moving said extra memory block forward one block at a time,until all of said number of memory blocks have been shifted one step in the address space.

2. (Original) The method as recited in claim 1, comprising the step of reversing the order of writing memory blocks in the address space, from said upgrade from a first version to a second version, to a second upgrade from said second version to a third version.

3. (Original) The method as recited in claim 2, comprising the steps of:

- said device, before receiving and executing a delta file, connecting with a server and communicating information related to a current address space status determining which

type of delta file is applicable for upgrading said software, dependent on the present location of said extra block in said address space, wherein a first delta file type is adapted to upgrade the software present in said memory blocks in one order in the address space, and a second delta file adapted to upgrade the software present in said memory blocks in a reversed order in the address space;

- downloading a delta file of the applicable delta file type from said server to said device; and
- upgrading said software using the applicable delta file.

4. (Currently amended) The method as recited in ~~any of the previous claims~~ claim 1, comprising the step of modifying a start address within said address space in a boot code for said software, dependent on the moving of data between said memory blocks.

5. (Currently amended) Computer program product, for use in a computer-controlled electronic device ~~(40)~~ for updating software present in a first version ~~(V1)~~ in the device, which device comprises means for receiving and executing a delta file ~~(D12)~~ for upgrading said first version to a second version ~~(V2)~~, which software is divided and stored in a number of memory blocks ~~(201-204)~~ defined in an address space ~~(211)~~ of a physical memory ~~(48)~~ in the device, said computer program product comprising executable computer program code devised to cause the device to perform the steps of:

- defining an extra memory block ~~(212)~~ associated with said number of memory blocks, initially being placed before a first memory block located at one end of the memory space;
- erasing said extra memory block;
- writing updated data for the first memory block on said extra memory block;
- moving the extra memory block one block forward; and
- processing all of said number of memory blocks one by one by erasing said extra memory block,

writing updated data for the memory block adjacent to and after the extra memory block on the extra memory block,

moving said extra memory block forward one block at a time,
until all of said number of memory blocks have been shifted one step in the address space.

6. (Currently amended) The computer program product as recited in claim 5, comprising executable computer program code devised to cause the device to perform the ~~steps of any of the previous claims 2-4~~ step of reversing the order of writing memory blocks in the address space, from said upgrade from a first version to a second version, to a second upgrade from said second version to a third version.

7. (Currently amended) Radio communication terminal ~~(40)~~ comprising a computer system ~~(47)~~ with associated computer code for updating software present in a first version ~~(V1)~~ in the terminal, and means ~~(46)~~ for receiving and executing a delta file ~~(D12)~~ for upgrading said first version to a second version ~~(V2)~~, which software is divided and stored in a number of memory blocks ~~(201-204)~~ defined in an address space ~~(211)~~ of a physical memory ~~(48)~~ in the terminal, characterised in that an extra memory block ~~(212)~~ associated with said number of memory blocks is defined in said address space, initially being placed before a first memory block located at one end of the memory space, said terminal further comprising means for erasing said extra memory block; data writing means for writing updated data for the first memory block on the extra memory block; means for moving the extra memory block one block forward; and means for processing all of said number of memory blocks one by one by
erasing said extra memory block,
writing updated data for the memory block adjacent to and after the extra memory block on the extra memory block,
moving said extra memory block forward one block at a time,

until all of said number of memory blocks have been shifted one step in the address space.

8. (Original) The radio communication terminal as recited in claim 7, comprising means for reversing the order of writing memory blocks in the address space, from said upgrade from a first version to a second version, to a second upgrade from said second version to a third version.

9. (Original) The radio communication terminal as recited in claim 8, wherein said device is arranged to connect with a server before receiving and executing a delta file, for communicating information related to a current address space status determining which type of delta file is applicable for upgrading said software, dependent on the present location of said extra block in said address space, wherein a first delta file type is adapted to upgrade the software present in said memory blocks in one order in the address space, and a second delta file adapted to upgrade the software present in said memory blocks in a reversed order in the address space, said terminal comprising means for downloading a delta file of the applicable delta file type from said server to said device, and means for upgrading said software using the applicable delta file.

10. (Currently amended) The radio communication terminal as recited in ~~any of the previous claims 7-9~~ claim 7, comprising means for modifying a start address within said address space in a boot code for said software, dependent on the moving of data between said memory blocks during writing as determined by said delta file.

11. (New) The computer program product as recited in claim 6, comprising executable computer program code devised to cause the device to perform the steps of:
- said device, before receiving and executing a delta file, connecting with a server and communicating information related to a current address space status determining which type of delta file is applicable for upgrading said software, dependent on the present

location of said extra block in said address space, wherein a first delta file type is adapted to upgrade the software present in said memory blocks in one order in the address space, and a second delta file adapted to upgrade the software present in said memory blocks in a reversed order in the address space;

- downloading a delta file of the applicable delta file type from said server to said device;

and

- upgrading said software using the applicable delta file.

12. (New) The computer program product as recited in claim 6, comprising executable computer program code devised to cause the device to perform the step of: modifying a start address within said address space in a boot code for said software, dependent on the moving of data between said memory blocks.